

## **REMARKS/ARGUMENTS**

### **Objections to Claims 2-11 and 18-20**

In the current Office Action, at paragraph 4 on page 2, the Examiner objected to Claims 2-11 and 18-20. The Examiner stated that the term "Claim" should be written as "claim."

In response, Applicants note that the Examiner has not cited any rule or law in support of this objection. Applicants submit that there is no legal basis for the Examiner to make this objection. Nonetheless, to expedite prosecution of this application on the merits, Applicants have revised Claims 2-11 and 18-20 in the manner requested by the Examiner.

None of these claim changes is believed to change the scope of these claims. Hence there is no doctrine of equivalence estoppel in making these changes. If the Examiner believes there has been a change in scope, the Examiner is requested to state the basis for their belief in the next office action.

### **Specification**

In the current Office Action, in paragraph 5 at the top of page 3, the Examiner objected to the specification for not providing antecedent basis for the limitation "writing a plurality of logs, at least one log."

In response, Applicants note that an Amendment dated November 23, 2007 already changed the specification at paragraph [0038] on page 13. Hence, the paragraph [0038] on page 13 of the specification as amended currently provides the antecedent basis which has been stated by the Examiner to be missing. Note that the Examiner has not raised a new matter objection. Besides, Applicants submit that there should be no new matter objection. Support for adding this language to paragraph [0038] on page 13 is found in originally-filed specification at page 20, claim 9. In view of this original support, no new matter is believed to be added, although the Examiner is requested to confirm this independently. Accordingly, Applicants respectfully request the Examiner to withdraw the objection to the specification.

The specification is now being amended above, at page 3 paragraph [0011] last sentence to correct a typographical error, namely the reference numeral 110 is replaced by 101. For support, see Applicants' FIG. 1 and also page 3 paragraph [0011] first sentence as originally filed. Accordingly, Applicants respectfully request entry of this amendment to the specification.

#### Amendments to Claim 1

Claim 1 is amended to recite "...generating an array update operation based on a query to update a relational database...". For support, for example, see the last sentence in paragraph [0011] at the top of page 4 of the original specification which states in pertinent part "operation 110 may be generated in the normal manner (e.g. via an SQL UPDATE query)."

Claim 1 is also amended to recite "... wherein said array update operation specifies a plurality of row-identifier and value pairs to update rows in a table of said relational database..." For support, see the first sentence in paragraph [0011] at the bottom of page 3 of the original specification which states in pertinent part "An array update operation ...specifies several (row-identifier, value) pairs ...updating multiple rows ... in a table of a relational database...."

Furthermore, Claim 1 is amended to recite "wherein said structure is located in main memory of said computer". For support, see the first sentence in paragraph [0013] on page 4 of the specification which states in pertinent part "Structure 113 (FIG. 1) can be implemented as any data structure in memory (e.g. main memory)."

Moreover, Claim 1 is also amended to recite that "each value comprises data to be updated in said row." Support for this amendment can be found throughout the application as originally filed, including, for example, page 1 paragraph [0001] ("a value V to be updated"), page 3 paragraph [0006] ("multiple rows ... are updated ... using values specified in the array update operation") and page 4 paragraph [0012] ("blocks known to contain rows to be updated").

Additionally, Claim 1 is further amended to recite "wherein several of said blocks are non-contiguous relative to one another in said storage device." Support for this

amendment can be found throughout the application as originally filed, including, for example, page 9 paragraph [0027] which states in pertinent part, "... one or more blocks that are not needing to be read but happen to be located between non-contiguous to-be-read blocks are also read."

#### Obviousness rejection of Claim 1

Claim 1 stands rejected under 35 U.S.C. §103 as being obvious over the teachings of US Patent 7,080,081 hereinafter Agarwal, in view of US Patent 5,517,631 hereinafter Machado. This rejection is respectfully requested to be withdrawn for several reasons as follows.

Firstly, note that Claim 1, as originally filed, required two different items, a "structure" and a "table". Claim 1 has now been amended to make more explicit a limitation that was previously inherent, namely that the claimed structure is in memory. This in-memory structure of Claim 1 is different from another item also being claimed in Claim 1, namely a table in a database.

Hence, Claim 1 requires an in-memory structure and an in-database table.

In the current Office Action, at the bottom of page 3, the Examiner stated "FIGS. 3-5 and 8 referred as structure". Agarwal's FIGS. 3-5 illustrate a "table" (see column 3, lines 13-22), and Agarwal's FIG. 8 illustrates a "block map" (see column 3, lines 29-30). The Examiner's remarks are unclear as to (a) Is Agarwal's "table" of FIGS. 3-5 analogized to Claim 1's in-memory structure or (b) Is Agarwal's "block map" of FIG. 8 analogized to Claim 1's in-memory structure. The Examiner's remarks are completely silent as to what item in Agarwal is analogized to Claim 1's in-database table. Due to these issues it is difficult for the undersigned to file a suitable response, and accordingly the next Office Action is respectfully requested to be made non-final.

Assume (a) above, that Agarwal's "table" is analogized to Claim 1's in-memory structure. If so, this rejection appears to be defective because the Examiner has not provided a rational basis to analogize anything in Agarwal against Claim 1's in-database table. Note that Agarwal's "table" appears insufficient to render obvious two distinct claim terms, an in-memory structure and an in-database table, as per Claim 1.

Assume (b) above, that Agarwal's "block map" being analogized to Claim 1's in-memory structure. If so, this rejection also appears to be defective because the Examiner has not provided a rational basis to analogize anything in Agarwal against Claim 1's in-database table. Hence, under either assumption (a) or (b) above, the Examiner's remarks appear to be insufficient to establish a prima facie case of obviousness. Accordingly this is a first reason for the patentability of Claim 1.

Secondly, Claim 1 has been amended to make more explicit another limitation that was previously inherent, namely generating an array update operation based on a query to update a relational database. Claim 1 further recites that the array update operation specifies row-identifier and value pairs, to update rows of a table.

In the Office Action on page 4 the Examiner admits that Agarwal states that "During operation, the query compiler 201 parses the input SQL commands and uses the code generator 202 to generate an execution plan. The parsed SQL commands are typically transformed into an internal representation and are then optimized". Agarwal appears to explain two internal representations in column 12 at lines 36-55.

The Examiner's remarks are unclear as to (c) Is Agarwal's "simple update" analogized to Claim 1's array update operation or (d) Is Agarwal's "Update of dimension column(s)" analogized to Claim 1's array update operation. The Examiner's remarks are completely silent as to which internal representation in Agarwal is analogized to Claim 1's array update operation. This is an additional reason why it is difficult for the undersigned to file a suitable response, and accordingly the next Office Action is respectfully requested to be made non-final.

Assume (c) above, that Agarwal's "Simple update" is analogized to Claim 1's array update operation. If so, this rejection appears to be defective because the Examiner has not provided a rational basis to replace the row update (which appears to be single row to the undersigned) as disclosed by Agarwal's column 12 line 39, with an operation to update multiple rows, as required in Claim 1.

Assume (d) above, that Agarwal's "Update of dimension column(s)" is analogized to Claim 1's array update operation. If so, this rejection also appears to be defective because the Examiner has again not provided a rational basis to replace a row insertion

and deletion (which appears to be of a single row) as disclosed by Agarwal's column 12 line 50-51 with an operation to update multiple rows, as required in Claim 1.

Accordingly this is a second reason for the patentability of Claim 1.

Thirdly, the Examiner admitted that Agarwal does not explicitly disclose "performing a single access operation without context switching." See the bottom paragraph on page 5 of the current Office Action. The Examiner then cited to Machado's 5 lines 33-40. The Examiner did not state at the bottom of page 5 of the Office Action, as to which column of Machado was being cited, which is another reason for making the next office action non-final. The Examiner's remarks are reproduced below:

However, Machado et al. discloses as the programmable data sequencer comprises a writeable control store including a random access memory area directly addressable by the programmed digital microcontroller for writing sequences of control patterns, there being most preferably a single sequence written for controlling states of the programmable data sequencer during both data read and data write operations to and from the disk surface and a buffer memory.

In the above-quoted text, Machado discloses that a single sequence is written for controlling the states of his sequencer during both read and write operations. Accordingly, Machado appears to support reading/writing of multiple blocks in a single sequence.

However, the above-quoted text of Machado appears to be completely silent as to the lack of "context switching" as recited in Claim 1. The Examiner failed to provide any rational basis to conclude that Machado discloses a method without context switching. Instead, Machado's method appears to operate independent of data within the blocks being read/written. Specifically, it appears that Machado does not care if a context switch happens or not, between two blocks in a sequence.

Moreover, Agarwal doesn't cure this defect in Machado, at least because the Examiner admitted that Agarwal does not explicitly disclose "performing a single access operation without context switching." More specifically, Applicants note that Agarwal

discloses each block being updated individually (see column 12, lines 37-55), and hence a context switch could happen between two blocks while performing Agarwal's method. Accordingly this is a third reason for the patentability of Claim 1.

A fourth issue is that the Examiner has not identified what specific item in Agarwal's disclosure is referred to as the "cache" of Claim 1.

A fifth issue is that Claim 1 now requires non-contiguous blocks, to distinguish over Agarwal's FIG. 8 (blocks 801-804 shown contiguous), and column 2 line 7.

If the Examiner continues to use Agarwal in a future Office Action, the Examiner is respectfully requested to explicitly identify what is referred to as the "structure" in Agarwal's patent, what is referred to as the "table" in Agarwal's patent, what is referred to as the "cache" in Agarwal's patent, and what is referred to as an "array update operation" in Agarwal's patent. Moreover, if the Examiner continues to use Machado or Agarwal in a future Office Action, the Examiner is respectfully requested to explicitly identify what is referred to as their lack of "context switch."

Thus, Applicants respectfully submit that Claim 1 is not obvious over the combined teachings of Agarwal and Machado. Reconsideration and withdrawal of this rejection is respectfully requested.

#### Rejection of Claims 2-11 and 19

Claims 2-11 and 19 depend from Claim 1 and are, therefore, likewise patentable for at least the above-discussed reasons in reference to Claim 1.

Claim 2 provides a further patentable distinction over Agarwal by reciting "sorting the block-identifiers, prior to retrieval of the blocks." The Examiner cited to Agarwal's column 5 lines 14-18 and column 6 lines 10-13 as disclosing Claim 2's sorting. Agarwal's text from these two columns is reproduced below:

#### Column 5 lines 14-18

The relevant data items are then retrieved and stored in the buffer manager 205 for reusability of the data. Typically, relational database management systems provide sequential table scan access as well as index-based access to tables.

Column 6 lines 10-12

While blocks are numbered sequentially starting from block 1 in the exemplary table shown herein, it should be appreciated that the blocks could be identified in numerous other ways.

Nothing in the above-quoted text by Agarwal appears to disclose sorting of block identifiers prior to retrieval of the blocks. The word "sort" is nowhere in the above-quoted text. At most Agarwal discloses what happens during a block's retrieval e.g. in a sequential table scan access or an index-based access. Claim 2 requires sorting "prior to" block retrieval. As the above-quoted text is the only text from Agarwal cited against Claim 2, the Examiner has failed to provide a rational basis to modify Agarwal to teach the "prior to" limitation in Claim 2.

Claim 19 further distinguishes over Agarwal by requiring Claim 2's sorting to be based on adjacency of blocks relative to one another which is not disclosed in the above-quoted text. Note that for Claim 19 as well, the above-quoted text is the only text from Agarwal cited against Claim 19. Hence, the Examiner has once again failed to make a prima facie case of obviousness.

Claim 3 provides a further distinction over Agarwal by reciting that the sorting is performed subsequent to storage of the block-identifiers in the structure. In explaining the rejection of Claim 3, the Examiner cited to Agarwal's column 2 lines 42-48 are reproduced below (while column 6 lines 10-12 are reproduced at the top of this page):

Column 2 lines 42-48

According to yet another aspect of the invention, processing a query further includes using information from either the individual block indexes or the composite index to obtain a list of block identifiers, and scanning blocks of the table for records. According to another aspect of the invention, processing a query includes the steps of scanning the entire table for records and using a record-based index to find records.

Nothing in the above-quoted text by Agarwal appears to disclose sorting "subsequent to" storage of block identifiers in the structure. Claim 3 is therefore patentable.

Claim 8 provides a further distinction over Agarwal by reciting that the structure comprises an array, and the array has a number of entries identical to the number of blocks that can be held in the cache. In explaining the rejection of Claim 8, the Examiner cited to Agarwal's column 2 lines 4-7 which are reproduced below:

Column 2 lines 4-7

A block index is created for each dimension of the table. According to another aspect of the invention, the blocks store information in a contiguous storage space.

Note that the Examiner also cited to Agarwal's FIG. 8. As noted above in reference to Claim 1 (from which Claim 8 depends), Agarwal's contiguous blocks are distinguished by reciting non-contiguous blocks in Claim 8. In this context, please note that Applicants' structure comprising the array is in memory, and distinct from Applicants' table which is in a database, and also distinct from Applicants' cache. Claim 8 is therefore patentable for this additional reason.

Claim 10 was rejected in the current Office Action for one or more reasons similar or identical to those discussed above in reference to Claim 1, and further in view of US Patent 6,321,234 granted to Debrunner. This rejection is respectfully traversed herewith. The Examiner-cited text, namely column 13 lines 53-67 and column 14 lines 33-40 appears to at most disclose that a log cache is flushed when unpinning a page. The Examiner has failed to provide a rational basis for extending Debrunner's disclosure to teach flushing an unpinned block to disk only when another block needs space in the cache occupied by the unpinned block as recited in Claim 10.

Claim 11 provides a further distinction over Agarwal by reciting that a plurality of file offsets are provided, one file offset for each block. In explaining the rejection of Claim 11, the Examiner cited to Agarwal's column 9 lines 40-51 which are reproduced below:



Column 2 lines 4-7

Each block may have a header, located in a first slot of the block's first page which stores a structure containing, among other possible things, a copy of the block status so that the block map can be re-created if necessary in case of deletion or corruption of the map, and a bit map covering the pages of the block, indicating which pages are empty (e.g., 0=empty, 1=nonempty, even if it contains only overflow or pointer records). Each block may also have a free space control record (FSCR) associated with it that could contain page offsets and approximations of free space per page. These FSCR's may be located on the first page of a block, and stored as the second record on this page, for example..

Nothing in the above-quoted text discloses any file offset, i.e. an offset within a file. Moreover, the Examiner has not provided a rational basis to replace Agarwal's page offsets with file offsets as required in Claim 11. Claim 11 is therefore patentable for this additional reason.

Rejection of Claims 6, 13, 15-17 and 20

Claims 6, 13, 15-17 and 20 recite one or more limitations that are supported by arguments for patentability that are similar to one or more of the arguments presented above in reference to Claim 1. Accordingly, these claims are also similarly patentable.

For the above reasons, Applicants respectfully request allowance of all pending claims. Please call the undersigned at (408) 378-7777 ext 113 in case of questions.

Via EXPRESS MAIL LABEL NO  
EM 170 359 416 US

Respectfully submitted,



Omkar K. Suryadevara  
Attorney for Applicant(s)  
Reg. No. 36,320